



PROIECT

Lanț de policlinici

Disciplina “Introducere in Baze de Date”

Realizat de:

Dragotoniu Corina Mădălina

Fazakas Edina

Ballai Levente

Facultatea de Automatică și Calculatoare

Anul II

Grupa 30223

Profesor coordinator: Ing. Cosmina Ivan

Cuprins

1. Introducere
2. Specificație proiect
3. Model de date
 - Tabele si attribute
 - Diagrama UML/EER
 - cod SQL
4. Aplicatia lantului de policlinici
 - mod de implementare
 - utilizare
5. Securitate si justificarea solutiei alese

1.Introducere

O bază de date este o colecție organizată de informații sau de date structurate, stocate electronic într-un computer. O bază de date este controlată, de regulă, de un [sistem de management al bazelor de date \(DBSM\)](#). Cumulat, datele, DBMS și aplicațiile asociate reprezintă un sistem de baze de date, denumit prescurtat bază de date.

Datele din cele mai obișnuite tipuri de baze de date sunt distribuite de regulă pe linii și coloane, în diferite tabele, pentru eficientizarea procesării și interogării datelor. Datele pot fi accesate, gestionate, modificate, actualizate, controlate și organizate cu ușurință. Majoritatea bazelor de date utilizează un limbaj structurat de interogare (SQL) pentru scrierea și interogarea datelor.

Proiectul nostru ce implementează un sistem informatic destinat gestiunii activităților dintr-un lanț de policlinici vine în ajutorul celor ce lucrează la acea policlinică, prin gestionarea mult mai ușoară a tuturor informațiilor.

Având în vedere sistemul actual, trăind într-o perioadă pandemică, acest sistem este esențial pentru buna desfășurare a tuturor activităților de la policlinică, orice personal necesar putând lucra de acasă (ex: receptioner, administrator, etc.). Vechiul mod de gestionare a datelor se dovedește a fi ineficient, putându-se greși ușor în preluarea și rescrierea unor informații. Este instabil atât din punct de vedere al securității, cât și din punctul de vedere al gestiunii timpului.

Cu toate acestea, proiectul acesta, dezvoltă o interfață complexă implementată prin intermediul limbajului JAVA ce permite accesarea diferitelor ferestre oferite de aplicație. Aceasta comunică în permanență cu baza de date SQL, în care sunt reținute toate datele.

În ceea ce urmează, vor fi prezentate modelul de date cu implementarea în SQL, aplicația lanțului de policlinici care prezintă atât implementarea cât și utilizarea, cât și securitatea și justificarea soluției alese.

2. Specificatie

Se dorește implementarea unui sistem informatic destinat gestiunii activităților dintr-un **lanț de policlinici**.

Lanțul de policlinici este format din mai multe unități medicale, fiecare fiind caracterizată prin denumire, adresă, descrierea serviciilor oferite și programul de funcționare, pentru fiecare zi a săptămânii.

Aplicația va trebui să utilizeze un sistem de gestiune pentru baze de date MySQL, iar interacțiunea cu acesta va fi realizată doar prin interfața grafică. Funcționalitățile pe care le va oferi programul vizează operații ce țin de gestiunea angajaților, serviciul financiar-contabil și administrarea operațiilor curente din cadrul policlinicii

(gestiunea pacienților programați, completarea unui raport medical, emiterea unui bon fiscal)¹.

Aplicația va putea fi accesată, pe baza unui proces de autentificare, de către mai multe tipuri de utilizatori, operând în departamentele resurse umane, financiar-contabil sau medical. Pentru fiecare tip de utilizator se vor reține informații precum CNP, nume, prenume, adresa, număr de telefon, email, cont IBAN, numărul de contract, data angajării, funcția deținută în cadrul lanțului de policlinici. Fiecare utilizator își va putea vizualiza datele personale imediat după ce va accesa sistemul informatic, fără a avea însă posibilitatea de a le modifica. Totodată, programul trebuie să ofere și o funcționalitate pentru deautentificare, prin care se revine la fereastra care solicită datele de acces, astfel încât și un alt utilizator să îl poată folosi ulterior, fără a fi necesară repornirea sa.

Utilizatorul de tip **administrator** poate adăuga, modifica și șterge informații în baza de date legate de utilizatori. De asemenea, va exista și un rol de **super-administrator** care poate opera inclusiv asupra utilizatorilor de tip administrator.

Pentru un utilizator de tip **angajat** se va reține salariul negociat și numărul de ore care trebuie realizat în fiecare lună. Funcțiile ce pot fi deținute în cadrul lanțului de policlinici sunt inspector resurse umane, expert financiar-contabil, receptioner, asistent medical și medic, corespunzătoare departamentelor de resurse umane, economic, respectiv medical.

Pentru un **asistent medical** se va reține suplimentar tipul² și gradul³.

Pentru un **medic** se va reține suplimentar specialitatea sau specialitățile în care își desfășoară activitatea, gradul⁴, codul de parafă, competențele pe care le deține pentru realizarea unor proceduri ce necesită acreditări speciale⁴, titlul științific⁵, postul didactic

⁶Totodată, fiecare medic are negociat un procent din serviciile medicale realizate care îi revine, adițional față de salariul negociat.

Lanțul de policlinici oferă pacienților un set de servicii medicale. O parte dintre acestea sunt disponibile pentru toate specialitățile (consultație, distinctă în funcție de gradul medicului care o realizează: specialist, primar, profesor / conferențiar), altele sunt specifice pentru fiecare specialitate în parte. Pentru un serviciu medical disponibil se va specifica specialitatea din care face parte, necesitatea existenței unei competențe a medicului care o efectuează, prețul asociat și durata (exprimată în minute).

Sistemul va fi format din mai multe module care vor putea fi accesate de angajați în funcție de drepturile pe care le dețin. Astfel, vor fi implementate un **modul pentru gestiunea resurselor umane** ce vizează gestiunea programului de lucru și al concediilor angajaților, un **modul pentru operații financiar-contabile** care determină profitul operațional ca diferență între venituri (sume încasate pentru serviciile medicale) și cheltuieli (plăți efectuate

către angajați aferente salariilor) și un **modul pentru gestiunea activităților operaționale** (programarea pacienților pentru servicii medicale și înregistrarea acestora în momentul în care se prezintă în clinica medicală, emiterea bonului fiscal de către recepționeri, completarea rapoartelor medicale de către asistenții medicali și medici).

Toate aceste module vor fi integrate în cadrul aceluiasi sistem informatic, sub forma unor meniuri care vor conține funcționalitățile pe care acestea le oferă, disponibilitatea lor fiind însă limitată și de permisiunile pe care le posedă utilizatorul autentificat la momentul respectiv de timp.

Drepturile de acces ale angajaților din diferite departamente la modulele sistemului sunt descrise în tabelul de mai jos:

Drepturile de acces ale angajaților aparținând unui departament
Legendă

- utilizatorul are drepturi de citire și de scriere
- utilizatorul are doar drepturi de citire / limitate la anumite funcționalități
- utilizatorul nu are nici un fel de drepturi

3.Model de date

Cod creare tabele:

```
drop database if exists Policlinica;
CREATE DATABASE IF NOT EXISTS Policlinica;
use Policlinica;
CREATE TABLE IF NOT EXISTS Angajat (
Id_angajat INT NOT NULL,
Functie VARCHAR(45) NOT NULL,
Salar_neg INT NOT NULL,
Nr_ore INT NOT NULL,
PRIMARY KEY (Id_angajat));

CREATE TABLE IF NOT EXISTS Asistent_medical (
Id_asis_medical INT NOT NULL,
Tip VARCHAR(20),
Grad VARCHAR(20),
Id_ca_utilizator INT NOT NULL,
id_de_la_medic int,
PRIMARY KEY (Id_asis_medical));
CREATE TABLE IF NOT EXISTS Medic (
Id_medic INT NOT NULL,
Specialitate VARCHAR(20) NOT NULL,
Grad VARCHAR(20) NOT NULL,
Cod_Parafa INT UNIQUE NOT NULL,
Competente_detinute VARCHAR(45) NOT NULL,
Titlu_Stiintific VARCHAR(20), -- poate sa nu aiba
Postul_Didactic VARCHAR(20), -- poate sa nu aiba
Procent_neg_servicii FLOAT(4,1) NOT NULL,
PRIMARY KEY(Id_medic),
Id_ca_utilizator INT NOT NULL);

CREATE TABLE IF NOT EXISTS Servicii (
Id_servicii INT NOT NULL,
Secialitate varchar(20),
Necesitate_competente VARCHAR(45),
Pret INT,
Durata INT,
echipament_necesar varchar(255),
PRIMARY KEY (Id_servicii));
drop table if exists Program;
```

```

CREATE TABLE IF NOT EXISTS Program(
id_program int primary key,
datapr date,
ora_deschidere int,
ora_inchidere int,
unitate int);
drop table if exists Unitati_Medicale;
CREATE TABLE IF NOT EXISTS Unitati_Medicale
(id_unitate_medicala int primary key,
denumire varchar(255),
adresa varchar(255));
drop table if exists cabinet;
CREATE TABLE IF NOT EXISTS cabinet(
id_cabinet int primary key,
id_unitate int,
id_medic int,
echipamente varchar(255));
drop table if exists Utilizatori;
CREATE TABLE IF NOT EXISTS Utilizatori(
id_utilizator int unique primary key, tip varchar(255),
parola varchar(50),
CNP int(13), nume varchar(255), prenume varchar(255),
adresa varchar(255), nr_telefon varchar(255),
mail varchar(255), cont varchar(255),
nr_contract int, data_angajarii date,
functie varchar(255), rol varchar(255) ,
vaccin varchar(10),
id_ca_angajat int);
drop table if exists Programari;
CREATE TABLE IF NOT EXISTS Programari(
id_programare int primary key,
id_pacient int,
id_serviciupr int,
id_serviciupr2 int,
id_serviciupr3 int,
ora_inceperii int,
ora_finalizarii int,
datapacient date, #yyyy-mm-dd
unitatea int,
cabinetul int,
medicul int);
drop table if exists ProgramAngajati;
CREATE TABLE IF NOT EXISTS ProgramAngajati(
id_prang int primary key,
id_ca_angajat int,
datapr date,
ora_incepere int,

```

```

ora_plecare int,
unitate int,
concediu bool);
drop table if exists Pacienti;
CREATE TABLE IF NOT EXISTS Pacienti(
id_pacient int primary key,
alergii varchar(255),
boli varchar(255),
nume varchar(30),
prenume varchar(50),
datanasterii date,
sexul char,
domiciliu varchar(255),
cetatenie varchar(30),
asigurare varchar(10));
drop table if exists IstoriceBonuri;
CREATE TABLE IF NOT EXISTS IstoriceBonuri(
id_bon int primary key,
id_de_la_programare int,
sumaplatita varchar(255));
drop table if exists raport_medical;
CREATE TABLE IF NOT EXISTS raport_medical(
id_fisa int primary key,
id_programare int,
istoric varchar(255),
simptome varchar(255),
investigatii varchar(255),
diagnostic varchar(255),
recomandari varchar(255),
validare int);
drop table if exists IstoriceSalarii;
CREATE TABLE IF NOT EXISTS IstoriceSalarii(
id_salariu int primary key,
id_angajat int,
data_aplicarii date,
salariu int);
Alter table IstoriceSalarii
add FOREIGN KEY (id_angajat) REFERENCES angajat(id_angajat);
Alter table Asistent_medical
add FOREIGN KEY (Id_ca_utilizator) REFERENCES utilizatori(id_utilizator),
add FOREIGN KEY (Id_de_la_medic) REFERENCES Medic(id_medic);
Alter table Medic
add FOREIGN KEY (Id_ca_utilizator) REFERENCES utilizatori(id_utilizator);
alter table raport_medical
-- add FOREIGN KEY (id_pacient) REFERENCES pacienti(id_pacient),
add FOREIGN KEY (id_programare) REFERENCES Programari(id_programare);
Alter table Utilizatori

```



```
add FOREIGN KEY (id_ca_angajat) REFERENCES Angajat(id_angajat);
Alter table Program
add FOREIGN KEY (unitate) references Unitati_Medicale(id_unitate_medicala);
Alter table Programari
add FOREIGN KEY (unitatea) references Unitati_Medicale(id_unitate_medicala),
add foreign key (id_serviciupr) references Servicii(id_servicii),
-- add foreign key (id_serviciupr2) references Servicii(id_servicii),
-- add foreign key (id_serviciupr3) references Servicii(id_servicii),
add foreign key (medicul) references Medic(Id_medic),
add foreign key (cabinetul) references cabinete(id_cabinet),
add FOREIGN KEY (id_pacient) REFERENCES pacienti(id_pacient);
Alter table ProgramAngajati
add FOREIGN KEY (unitate) references Unitati_Medicale(id_unitate_medicala),
add FOREIGN KEY (id_ca_angajat) references Angajat(Id_angajat);
alter table cabinete
add foreign key(id_unitate) references Unitati_Medicale(id_unitate_medicala),
add foreign key(id_medic) references Medic(Id_medic);
Alter table Istoriconuri
add FOREIGN KEY (id_de_la_programare) REFERENCES Programari(id_programare);
```

View-uri folosite:

Pentru a afisa orarul programarilor am folosit view-ul:

```
DROP VIEW IF EXISTS Orar_programari;

CREATE VIEW Orar_Programari AS
SELECT CONCAT(P.num, ' ', P.preume) AS 'Pacient', ora_inceperii, ora_finalizarii,
CONCAT(U.num, ' ', U.preume) AS 'medic'
FROM programari Pr, medic M, pacienti P, utilizatori U
WHERE id_med = medicul AND Pr.id_pacient = P.id_pacient
AND Id_ca_utilizator = id_utilizator;
```

Pentru a afisa programul angajatilor am folosit view-ul:

```
DROP VIEW IF EXISTS Program_Angajat;

CREATE VIEW Program_Angajat AS
SELECT datapr AS 'Data', CONCAT(Pr.ora_incepere, ' - ', Pr.ora_plecure) AS 'Program',
U.denumire AS 'Unitate medicala', U.adresa, utilizatori.num, id_angajat
FROM programangajati Pr, angajat A, unitati_medicale U, utilizatori
WHERE id_unitate_medicala = unitate
AND id_angajat = Pr.id_ca_angajat AND id_angajat = utilizatori.id_ca_angajat;
```

Proceduri folosite:

Pentru a verifica daca un angajat e in concediu:

```
DELIMITER //
DROP PROCEDURE IF EXISTS verificare;
CREATE PROCEDURE verificare(n varchar(255), p varchar(255), t varchar(255), datav date) BEGIN
SELECT concediu FROM programangajati Pr, utilizatori U, angajat A
WHERE U.id_ca_angajat = A.id_angajat AND Pr.id_ca_angajat = A.id_angajat
AND U.num = n AND U.preume = p AND tip = t AND datapr = datav;
```

```
END
// DELIMITER
```

Pentru a verifica daca e in concediu intr-o anumita data:

```
DELIMITER //
DROP PROCEDURE IF EXISTS verificareAconcediu;
CREATE PROCEDURE verificareAconcediu(id int, p varchar(50), datav date)
BEGIN
SELECT concediu FROM programangajati Pr, utilizatori U, angajat A
WHERE U.id_ca_angajat = A.id_angajat AND Pr.id_ca_angajat = A.id_angajat
AND A.id_angajat = id AND U.parola = p AND datapr = datav;
END
// DELIMITER
```

Pentru a verifica daca o unitate are program in ziua respectiva:

```
DELIMITER //
DROP PROCEDURE IF EXISTS CautareProgramAzi;
CREATE PROCEDURE CautareProgramAzi(n varchar(255), p varchar(255), t varchar(255)) BEGIN
SELECT ora_incepere AS 'Ora incepere', ora_plecure AS 'Ora plecure', unit.denumire AS 'Unitate'
FROM Utilizatori U, Angajat A, Programangajati Pr, Unitati_medicale unit
WHERE U.id_ca_angajat = A.id_angajat AND Pr.id_ca_angajat = A.id_angajat AND Pr.unitate =
unit.id_unitate_medicala
AND U.num = n AND U.prenume = p AND tip = t AND datapr = current_date();
END
// DELIMITER
```

Pentru a modifica programul unui angajat:

```
DELIMITER //
DROP PROCEDURE IF EXISTS ModificareProgramAngajat;
CREATE PROCEDURE ModificareProgramAngajat(n varchar(255), p varchar(255), t varchar(255),
datav date, inceput int , final int)
BEGIN
UPDATE programangajati Pr, angajat A, utilizatori U
SET ora_incepere = inceput, ora_plecure = final
WHERE U.id_ca_angajat = A.id_angajat AND Pr.id_ca_angajat = A.id_angajat
AND U.num = n AND U.prenume = p AND tip = t AND Pr.datapr = datav;
END
// DELIMITER
```

Pentru a adauga o zi de concediu la un anagajat:

```
DELIMITER //
-- DROP PROCEDURE IF EXISTS AdaugareConcediu;
```

```

CREATE PROCEDURE AdaugareConcediu(n varchar(255), p varchar(255), t varchar(255), inceput date
, final date)
BEGIN
DECLARE c int;
SELECT COUNT(id_programare) INTO c
FROM programari P, unitati_medicale UM, programangajati Pr, angajat A, utilizatori U
WHERE U.id_ca_angajat = A.id_angajat AND id_angajat = Pr.id_ca_angajat AND id_unitate_medicala =
P.unitatea
AND id_unitate_medicala = Pr.unitate AND nume = n AND prenume = p AND tip = t;

IF(c = 0) then UPDATE programangajati Pr, angajat A,
utilizatori U
SET ora_incepere = null, ora_plecure = null, concediu = 1
WHERE U.id_ca_angajat = A.id_angajat AND Pr.id_ca_angajat = A.id_angajat
AND U.nume = n AND U.prenume = p AND tip = t AND datapr BETWEEN inceput AND final; end
if;
END

```

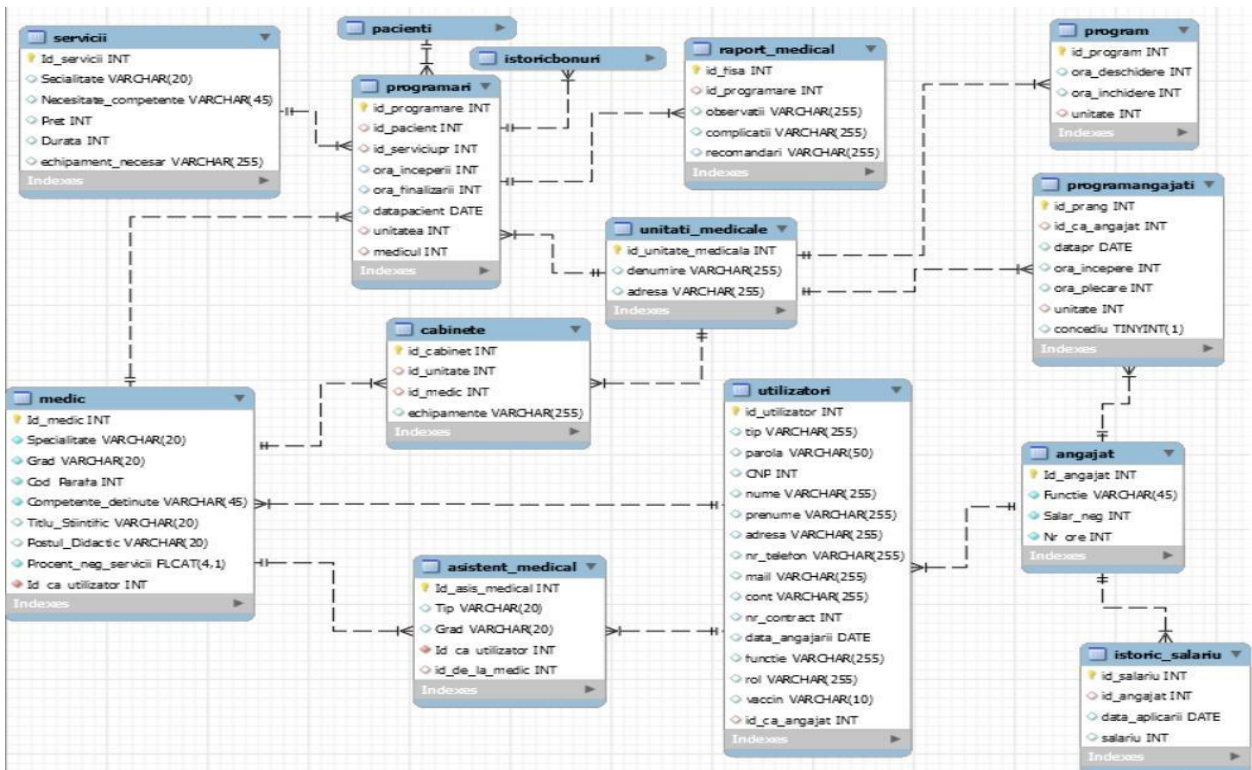
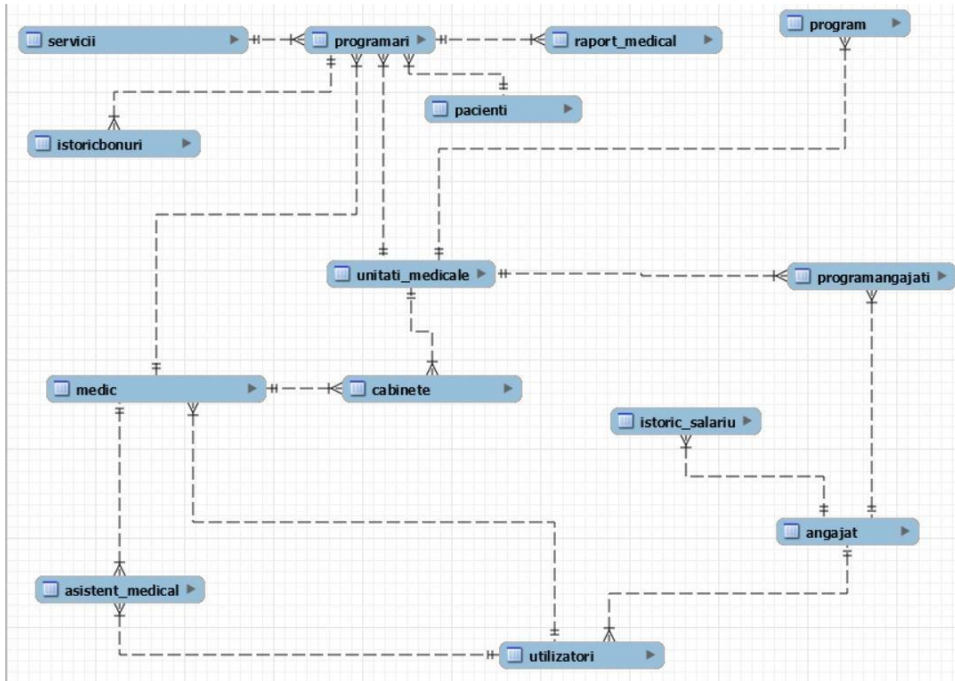
Pentru a verifica daca intervalul orar introdus pentru o consultatie e egal cu intervalul orar necesar unei consultatii:

```

// DELIMITER
DELIMITER $$
CREATE PROCEDURE verificare_ore1( s1 int, s2 int, s3 int, oi int , orf int )
BEGIN
    select if(((s1+s2+s3)/60 =orf-oi),1,0) as ifc ;
END $$
DELIMITER ;

```

Diagrama UML:



4. Implementarea aplicatiei:

1. Pagina principala



Acesta este prima pagina care se deschide atunci cand utilizatorul acceseaza aplicatia. Si acestuia i se ofera posibilitatea de a intra la sectiunea de gestiune a resurselor umane, de operatii financiar contabile, de gestiune a activitatilor operationale, de administrator, super-administrator, respectiv vaccine dand click pe butoanele disponibile pe interfata.

2. Administrator

Politicină

ID Utilizator:

Tip:

Parolă:

CNP:

Nume:

Prenume:

Adresa:

Nr. telefon:

E-mail:

Cont IBAN:

Nr. contract:

Data angajării:

Funcție:

Rol:

ID ca angajat:

Vizualizare Utilizatori

Inserare in tabel Update tabel Ștergere din tabel

Căutare după ID: Căutare după nume:

Căutare Delogare

Pe aceasta pagina administratorului i se permite sa adauge, modifice, stearga utilizatori din tabelul Utilizatori, insa doar daca acestia nu sunt de tip administrator sau superadministrator.

Acesta trebuie sa completeze casutele aflate in stanga cu datele care trebuie introduce in tabel, apoi sa apese butonul

Inserare/Update pentru a face operatiile respective.

Pentru a sterge e de ajuns sa completeze doar id-ul utilizatorului.

Acesta poate sa caute utilizatori in tabela in functie de id-ul lor sau dupa nume, selectand din ComboBox id-ul sau completand numele in casuta de text.

3. Super-Administrator

Policlina

ID Utilizator:

Tip:

Parolă:

CNP:

Nume:

Prenume:

Adresa:

Nr. telefon:

E-mail:

Cont IBAN:

Nr. contract:

Data angajării:

Funcție:

Rol:

ID ca angajat:

Vizualizare Utilizatori

Inserare in tabel

Update tabel

Ștergere din tabel

Căutare după ID: 1

Căutare după nume:

Căutare

Delogare

Pe aceasta pagina super-administratorului i se permite sa adauge, modifice, stearga utilizatori din tabelul Utilizatori.

Acesta trebuie sa completeze casutele aflate in stanga cu datele care trebuie introduce in tabel, apoi sa apese butonul Inserare/Update pentru a face operatiile respective.

Pentru a sterge e de ajuns sa completeze doar id-ul utilizatorului.

Acesta poate sa caute utilizatori in tabela in functie de id-ul lor sau dupa nume, selectand din ComboBox id-ul sau completand numele in casuta de text.

Modulul I. Resurse umane

În acest modul se pot vizualiza informații despre propria persoană în ceea ce privește orarul pentru angajați (receptioner, medic, asistent), pentru exectorul financiar există secțiunea de program și concediu al tuturor angajaților secțiuni asemănătoare cu cele ale angajaților de tip resurse umane; pentru ei se adaugă posibilitatea de modificare.



The image shows a login window titled "LOGARE CA:" with a yellow background. It contains three blue buttons with white text: "Administrator resurse umane", "Administrator economic", and "Angajat*". At the bottom left, there is a small text note: "*Angajat al unității medicale: receptioner, medic, asistent". At the bottom right, there is a blue button with white text: "Înapoi". The window has a standard Windows title bar with a red cross icon and the text "Policlinică".

Prima fereastră este cea de alegere, fiecare utilizator își selectează categoria potrivită.

Pentru logarea ca administrator resurse umane, dacă datele au fost introduse bine, va apărea fereastra pentru care am atașat imaginea din stânga. Butonul de „Afișare angajați” listează în tabel datele posibile de introducere pentru a începe căutarea. „Căutare angajat*” va afișa programul din ziua curentă. Pentru a căuta o anumită dată calendaristică se poate folosi „Căutare zi” după ce e introdusă data în câmpul liber „Data”. La fel se poate proceda și pentru căutarea unei perioade. Modificare se face după apăsarea butonului dacă sunt completate toate câmpurile necesare.

Pentru „Concediu” vine in ajutor butonul de afisare angajati, daca se apasa pe o linie din tabelul angajat afisat datele din stanga tabelului se vor completa automat cu cele din randul ales.

Este posibila afisarea concediilor din tot istoricul angajatului si modificarea programului in vederea adaugarii concediului. In spatele butonului de

„Modificare” este apelata o procedura `call AdaugareConcediu`. Concediul nu poate fi adaugat daca angajatul (medic sau asistent) are deja programari pentru data respectiva.

```
CREATE PROCEDURE AdaugareConcediu(n
varchar(255), p varchar(255), t varchar(255), inceput date , final date)
BEGIN
DECLARE c int;
SELECT COUNT(id_programare) INTO c
FROM programari P, unitati_medicale UM, programangajati Pr, angajat A, utilizatori U
WHERE U.id_ca_angajat = A.id_angajat AND id_angajat = Pr.id_ca_angajat AND id_unitate_medicala
= P.unitatea
AND id_unitate_medicala = Pr.unitate AND nume = n AND prenume = p AND tip = t;

IF(c = 0)
then UPDATE programangajati Pr, angajat A, utilizatori U
SET ora_incepere = null, ora_plecure = null, concediu = 1
WHERE U.id_ca_angajat = A.id_angajat AND Pr.id_ca_angajat = A.id_angajat
AND U.nume = n AND U.prenume = p AND tip = t AND datapr BETWEEN inceput AND final; end
if;
END
```

Politiclinică

Căutare angajat Orar săptămânal Concediu Personalizare servicii medicale

Introduceți datele necesare personalizării unui serviciu medical:

ID:

Specialitate:

Competențe:

Preț:

Durată:

Echipament necesar:

Id	Specialitatea	Competențe	Preț	Durata	Echipament
1	cardiologie	specialist	100	60	ecograf
2	cardiologie	specialist	200	60	holter
3	neurologie	specialist	100	60	CT
4	pediatrie	specialist	100	60	Stetoscop si tensi.
5	ortopedie	specialist	100	60	punere ghips

Pentru personalizarea de servicii medicale, tab accesibil doar departamentului resurse umane, se pot adauga date, pentru a completa mai usor se introduc automat cand este apasata o linie din tabel, se modifica informatiile din campurile dorite, iar butonul „Adauga” va duce la afisarea unui mesaj daca au fost introduse cu succes sau nu.

Vă rugăm introduceți datele:

ID:

Parolă:

Data	Program	Unitatea
------	---------	----------

Introduceți o dată calendaristică pentru a afișa programul:

Sau introduceți o perioadă:

-

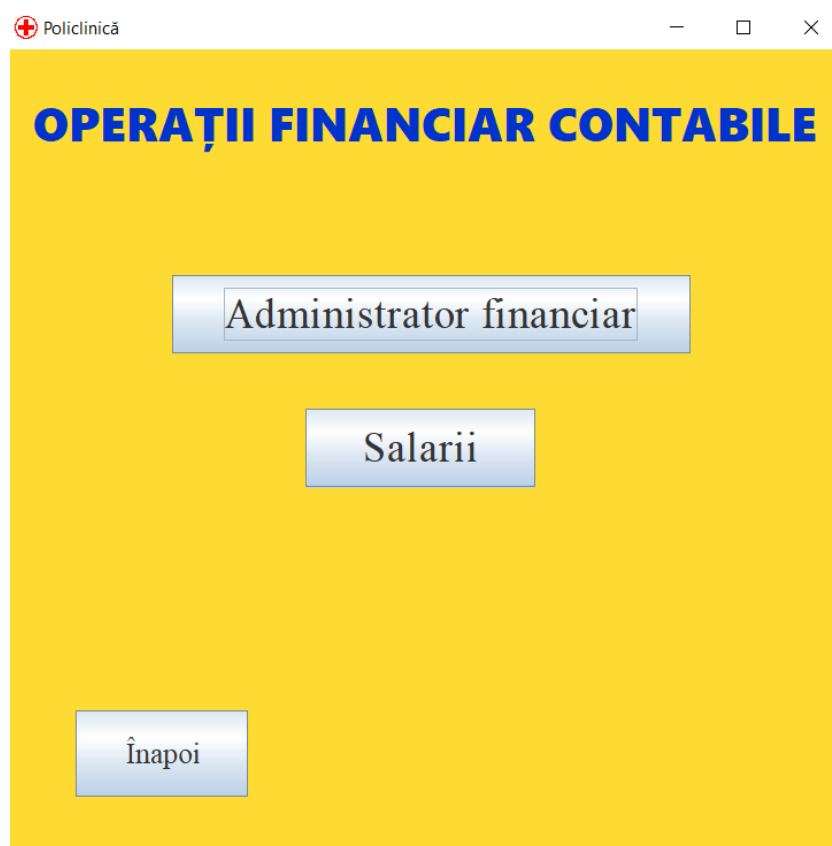
Angajatii nu au nevoie de logare inaintede a vizualiza fereastra, deoarece pentru a cauta informatii legate de program sau concediu trebuie sa completeze aici idul si parola.

Pot cauta personalizat programul, cu alte cuvinte pot cauta o anume zi calendaristica sau perioada.

Inclusiv pentru o zi pot vizualiza programarile.

Modulul II. Operatii financiar contabile

În acest modul, un utilizator va putea să își verifice atât salariul, cât și istoricul salariilor. Mai mult, un medic va putea să verifice și profitul generat în luna actuală, defalcat pe venituri și cheltuieli. La selectarea opțiunii “Operatii dinanciar contabile”, se va deschide următoarea fereastră:



1. Administrator financiar

Înainte de a i se deschide fereastra de unde poate selecta orice dorește să vadă, acesta trebuie să se conecteze. Dacă este administrator financiar (tip='economic'), se va deschide următoarea fereastră de unde poate monitoriza fiecare angajat în parte. În caz contrar, se va afișa un mesaj "Wrong id and password".



The image shows a Windows-style window titled "Policlinică" with standard minimize, maximize, and close buttons. The window has a yellow background and the title "Administrator financiar" in large blue letters. It contains two input fields: "ID:" and "Parolă:". Below these fields are two buttons: "Înapoi" (Back) and "Logare" (Login).

Dacă serverul detectează că cel care dorește să acceseze secțiunea nu este administratorul financiar, se deschide o fereastră pentru administratorul financiar care se deschide după ce a reușit conectarea. Prezintă 5 butoane, fiecare în parte lăsându-l pe administratorul financiar să observe profitul generat de fiecare medic în parte, pe o anumită locație sau specialitate medicală din luna precedentă. De asemenea, se poate vedea și profitul total adus

de lantul de policlinici, dar si profitul lunar adus de asemenea de intreg lantul de policlinici. Profitul se poate vizualiza doar in cazul in care exista activitate in luna precedenta.

adresa	venituri	cheltuieli	profit
Suceava	400.0	3280.0	-2880.0
Cluj	400.0	3200.0	-2800.0
Oradea	600.0	3513.0	-2913.0
Brasov	200.0	3162.0	-2962.0

Exemplu:

□ Selectarea profitului generat de medici pe unitate medicala

```
SELECT unitati_medicale.adresa, SUM(sumaplatita) as venituri,
IF(COUNT(programari.medicul),3000+SUM(medic.Procent_neg_servicii/100 * sumaplatita),3000) as
cheltuieli, SUM(sumaplatita)-3000-SUM(medic.Procent_neg_servicii/100 * sumaplatita) as profit
from angajat, utilizatori, istoricbonuri,medic,programari, unitati_medicale where
utilizatori.Id_ca_angajat=angajat.Id_angajat and programari.medicul=medic.Id_medic and
medic.Id_ca_utilizator=utilizatori.id_utilizator
and istoricbonuri.id_de_la_programare=programari.id_programare and
programari.unitatea=unitati_medicale.id_unitate_medicala
and MONTH(programari.datapacient)=MONTH(CURRENT_DATE - INTERVAL 1 MONTH) group
by unitati_medicale.adresa;
```

adresa	venituri	cheltuieli	profit
Suceava	200.0	3140.0	-2940.0
Cluj	400.0	3200.0	-2800.0
Oradea	600.0	3513.0	-2913.0
Brasov	200.0	3162.0	-2962.0

Cea de a doua fila de la fereastra administratorului financiar ne ofera informatii despre salariile angajatilor din luna precedenta.

Salariile medicilor sunt calculate in functie de ce programari au efectuat. In cazul in care nu au activitate pe acea luna, ei vor avea un salariu (minim) de 3000 lei.

Butonul “Alte salarii” ne ofera informatii despre salariile altor angajati care nu sunt medici, asistenti sau receptioneri.

Exemplu:

□ afisarea salariilor medicilor:

```
SELECT nume,prenume,
3000+SUM(medic.Procent_neg_servicii/100 * sumaplatita) as salariu
from istoricbonuri,utilizatori,medic,programari WHERE
medic.id_medic=programari.medicul
and programari.id_programare=istoricbonuri.id_de_la_programare
and utilizatori.id_utilizator=medic.id_ca_utilizator
```


and MONTH(programari.datapacient)=MONTH(CURRENT_DATE - INTERVAL 1 MONTH) group by medic.id_medic;

nume	prenume	salariu
Ionescu	Matei	3140.0
Apostol	Maria	3200.0
Vasilescu	Ioana	3513.0
Ivascu	Andreea	3162.0

2. Salarii

Cea de a doua optiune din fereastra initiala “Operatii financiar contabile”, ofera informatii despre angajatul ce o acceseaza pe luna precedenta. Pentru a putea selecta una din optiunile afisate, mai intai trebuie sa se inregistreze, pentru a i se putea furniza informatii doar despre el.

Afișează salariu actual

Istoric salarii

Profit generat

ID:

Parolă:

luna	salariu
12	3280.0

Înapoi

Optiunea “Profit generat” este disponibila doar celor care sunt medici.

Istoricul salariilor pentru medici se arata doar in momentul in care un medic are activitate pe luna respective, altfel se considera salariul de baza 3000 lei. Pentru restul personalului, cu un salariu fix, se va afisa data de la care a fost aplicat un salariu si salariul respective.

Exemplu:

medic

➤ salariul actual

```
SELECT MONTH(CURRENT_DATE - INTERVAL 1 MONTH) as luna,  
IF(COUNT(programari.medicul),3000+SUM(medic.Procent_neg_servicii/100 * sumaplatita),3000) as salariu  
FROM istoricbonuri, programari,medic  
WHERE MONTH(programari.datapacient)=MONTH(CURRENT_DATE - INTERVAL 1 MONTH)  
AND istoricbonuri.id_de_la_programare=programari.id_programare  
and programari.medicul=medic.id_medic  
and medic.id_ca_utilizator=user.getText();
```

➤ istoric salarii

```
SELECT m.luna, IF(COUNT(programari.medicul),3000+SUM(medic.Procent_neg_servicii/100 *  
sumaplatita),3000) as salariu FROM (SELECT 'January' AS luna  
UNION SELECT 'February' AS luna  
UNION SELECT 'March' AS luna  
UNION SELECT 'April' AS luna  
UNION SELECT 'May' AS luna  
UNION SELECT 'June' AS luna  
UNION SELECT 'July' AS luna  
UNION SELECT 'August' AS luna  
UNION SELECT 'September' AS luna  
UNION SELECT 'October' AS luna  
UNION SELECT 'November' AS luna  
UNION SELECT 'December' AS luna) AS m, istoricbonuri, programari,medic  
WHERE istoricbonuri.id_de_la_programare=programari.id_programare and  
programari.medicul=medic.id_medic and medic.id_ca_utilizator= + user.getText()  
+  
and m.luna=monthname(programari.datapacient)  
group by m.luna;
```

Afișează salariu actual

Istoric salarii

Profit generat

ID:

Parolă:

luna	salariu
December	3280.0
November	3140.0
July	3140.0

Înapoi

➤ profit generat

```

SELECT MONTH(CURDATE()) as luna,
IF(COUNT(programari.medicul),SUM(((100-medic.Procent_neg_servicii)/100 * sumaplatita)-3000,-3000) as profit
FROM istoricbonuri, programari,medic
WHERE MONTH(programari.datapacient)=MONTH(CURRENT_DATE - INTERVAL 1 MONTH)
AND
istoricbonuri.id_de_la_programare=programari.id_programare
and programari.medicul=medic.id_medic and
medic.id_ca_utilizator=user.getText();

```

Policlinică

Afișează salariu actual

ID:

Parolă:

Istoric salarii

Profit generat

luna	profit
1	-2880.0

Înapoi

orice angajat in afara medicilor

➤ salariu actual

```
SELECT nume, prenume, salar_neg*nr_ore*(28-SUM(IF(programangajati.concediu=1,1,0))) as salariu FROM
Angajat, utilizatori, programangajati
WHERE angajat.id_angajat=utilizatori.id_ca_angajat
AND MONTH(programangajati.datapr)=MONTH(CURRENT_DATE - INTERVAL 1 MONTH)
AND utilizatori.id_utilizator= user.getText();
```

➤ istoric angajat

```
SELECT salariu, data_aplicarii
FROM Istoric_Salariu, angajat, utilizatori
WHERE angajat.id_angajat=Istoric_Salariu.id_angajat
AND angajat.id_angajat=utilizatori.id_ca_angajat
AND utilizatori.id_utilizator=+ user.getText() +;
```

Modul III: Gestiunea activitatilor operationale

LOGARE CA:

Receptioner

Medic

Asistent medical

Înapoi

AUTENTIFICARE RECEPTIONER

ID:

Parolă:

Înapoi

Logare

Atunci cand acceseaza aceasta pagina, utilizatorul trebuie sa isi aleaga mai intai drept ce se logheza, apasand in continuare butoanele corespunzatoare, deschizandu-se mai apoi o pagina in care trebuie sa-si introduca id-ul si parola pentru a putea continua mai departe. Daca a introdus datele corecte i se deschide urmatoarea pagina corespunzatoare, altfel primeste o atentionare „Wrong id or password”.

Daca s-a autentificat drept asistent medical:

ID fișă:

ID programare:

Istoric:

Simptome:

Investigații:

Diagnostic:

Recomandări:

Validare:

Adăugare

Modificare

Vizualizare raporte medicale

id_fisa	id_programare	istoric	simptome	investigati	diagnostic	recomandari	validare
1	1	programare holter	anemie	consult	bla	mancat mai sanat	1
2	2	programare holter	roseata	consult	bla	mancat mai sanat	2
3	3	programare holter	insuficienta cardia	consult	bla	mancat mai sanat	3
4	4	programare holter	hipertensiune	consult	bla	mancat mai sanat	4
5	5	programare holter	anemie	consult	bla	mancat mai sanat	5
6	6	programare holter	roseata	consult	bla	mancat mai sanat	6
7	7	programare holter	insuficienta cardia	consult	bla	mancat mai sanat	7
8	8	programare holter	hipertensiune	consult	bla	mancat mai sanat	8
9	9	programare holter	anemie	consult	bla	mancat mai sanat	1
10	10	programare holter	roseata	consult	bla	mancat mai sanat	2
11	11	programare holter	insuficienta cardia	consult	bla	mancat mai sanat	3
12	12	programare holter	hipertensiune	consult	bla	mancat mai sanat	4
13	13	programare holter	anemie	consult	bla	mancat mai sanat	5
14	14	programare holter	roseata	consult	bla	mancat mai sanat	6
15	15	programare holter	insuficienta cardia	consult	bla	mancat mai sanat	7
16	16	programare holter	hipertensiune	consult	bla	mancat mai sanat	8
17	14	b	b	b	b	b	17
18	13	b	b	diffriesklovu	b	in	-1
19	13	b	b	b	b	b	1

Căutare după ID:

Delogare

I se deschide fereastra de sus, astfel ca acesta poate vizualiza rapoartele medicale, poate adauga si modifica rapoarte medicale, completand casutele text de langa tabel si poate cauta rapoare dupa id. Acesta mai poate accesa datele din tabel care se copiaza automat in casute in momentul cand acesta da click pe un rand din tabel.

Daca s-a autentificat ca medic:

The image displays three screenshots of a medical software interface, likely for a doctor's office. The interface is divided into three main panels, each with a yellow header bar.

Panel 1 (Top Left): This panel is titled "Pacienti" (Patients). It contains a search form on the left with fields for "ID pacient:", "Alergi:", "Boli:", "Nume:", "Prenume:", "Data nașterii:", "Sexul:", "Domiciliu:", "Cetățenie:", and "Asigurare:". To the right of the search form is a table listing patients with columns: "id_pacient", "alergi", "boli", "nume", "prenume", "data_nastere", "sex", "domiciliu", "cetatenie", and "asigurare". Below the table are search filters: "Căutare pacient după ID:", "Căutare pacient după nume:", "Căutare pacient după prenume:", and "Căutare pacient după ID-ul programării:". A "Vizualizare tuturor pacienților" button is located below the filters.

Panel 2 (Top Right): This panel is titled "Programări pacienți" (Patient appointments). It contains a search form on the left with fields for "ID programare:", "ID pacient:", "ID serviciu1:", "ID serviciu2:", "ID serviciu3:", "Ora începerii:", "Ora finalizării:", "Data:", "Unitates medicală:", and "Medic:". To the right of the search form is a table listing appointments with columns: "id_program", "id_pacient", "id_serviciu1", "id_serviciu2", "id_serviciu3", "ora_inceperii", "ora_finalizarii", "data_pacient", "unitate_medicala", and "medic". Below the table are search filters: "Căutare programare după ID:", "Căutare programare după nume:", "Căutare programare după prenume:", and "Căutare programare după ID-ul programării:". A "Vizualizare programări" button is located below the filters.

Panel 3 (Bottom): This panel is titled "Rapoarte Medicale" (Medical Reports). It contains a search form on the left with fields for "ID fișă:", "ID programare:", "Istoric:", "Simptome:", "Investigații:", "Diagnostic:", "Recomandări:", and "Validare:". To the right of the search form is a table listing medical reports with columns: "id_fisa", "id_programare", "data", "simptome", "investigatii", "diagnostic", "recomandari", and "validare". Below the table are search filters: "Căutare raport medical cu ID-ul fișei:", "Căutare raport medical cu ID-ul programării:", and "Căutare raport medical cu ID-ul pacientului:". A "Vizualizare rapoarte medicale" button is located below the filters.

Pe primul panel, cel cu Pacienti, acesta poate vizualiza informatiile despre pacienti, dar poate si cauta pacienti in functie de id, nume, prenume, programare.

Pe al doilea panel, cel cu Programari Pacienti, acesta poate vedea programările din ziua respectiva si sa caute programările in functie de anumite date in cazul in care are nevoie de una anume.

Pe al treilea, acesta poate vizualiza, adauga si valida rapoartele medicale, dar poate si sa caute in functie de id-ul fișei, id-ul programarii sau dupa id-ul pacientului.

Pe ultimul panel, acesta poate vizualiza, adauga sau sterge un serviciu. Utilizatorul poate accesa mai usor datele din tabele, dand click pe un anumit rand din tabel, iar datele din acel rand vor aparea in casutele din stanga tabelului.

Daca s-a autentificat ca receptioner:

Pe primul panel acesta poate vizualiza, adauga, sterge pacientii, poate mofifica dateleleor, dar poate si cauta un anume pacient dupa nume sau prenume.

Vizualizeaza atat programarile, medicii, serviciile, cabinetele si unitatile medicale pentru a-l ajuta sa completeze datele in casutele din stanga tabelului. Pentru a adauga/modifica/sterge o programare, aceasta trebuie sa fie verificata. Dand click pe butonul verificare, se verifica, daca in acelasi timp are loc o alta programare, la acelasi cabinet de la aceasi unitate si cu acelasi doctor, intre programari trebuind si sa existe o pauza de o ora de odihna pentru medici. Poate cauta eventual si programri in functie de cateva date, pentru a-i siimplifica alegerea datelor pe care vrea sa le introduca in tabel. Daca o programare nu este verificata si validata in urma verificarii aceasta nu poate fi modificata, adaugata sau stearsa.

Policlina

Pacienti | Programari | Serv Fisical | Programare speciala

ID pacient: 12

Alergii:

Boli:

Nume:

Prenume:

Data nastere:

Sexul:

Domiciliu:

Cetățenie:

Asigurare:

id_pacient	alergii	boli	nume	prenume	data_nasterii	sexul	domiciliu	cetatenie	asigurare
21			Chiriac	Valentin	1975-02-05	ms	Comana	romana	CA
22			Bucurariu	Adrian	1985-08-18	ms	Brasov	romana	CA
23			Combar	Alina	2000-12-05	f	Chiriac	romana	CA
24			Bucurariu	Adrian	1985-08-18	ms	Chiriac	romana	CA
25			Combar	Adrian	1977-05-12	f	Brasov	romana	CA
26			Bucurariu	Adrian	1985-08-17	ms	Brasov	romana	CA
27			Combar	Adrian	1977-05-12	f	Chiriac	romana	CA
28			Combar	Adrian	2001-02-13	ms	Brasov	romana	CA
29			Combar	Adrian	1988-08-17	ms	Chiriac	romana	CA
30			Combar	Adrian	2001-02-13	ms	Brasov	romana	CA
31			Combar	Adrian	2001-02-13	f	Chiriac	romana	CA

Vizualizare Pacienti Insetare Pacient

Modificare date de la un pacient Stergere Pacient

Căutare după nume:

Căutare după prenume:

Delogare

Policlina

Pacienti | Programari | Serv Fisical | Programare speciala

ID bon: 10

ID de la programare:

Suma platita:

id_program	id_pacient	id_serviciu	id_serviciu	ora_inceperii	ora_finalizarii	data_pacient	unitatea	cabinetul	medicul
1	21	1	0	0	0	2021-12-12	1	1	1
2	22	2	0	0	0	2021-12-13	2	2	2
3	23	3	0	0	0	2021-12-14	3	3	3
4	24	4	0	0	0	2021-12-15	4	4	4
5	25	5	0	0	0	2021-12-16	5	5	5
6	26	6	0	0	0	2021-12-17	6	6	6
7	27	7	0	0	0	2021-12-18	7	7	7
8	28	8	0	0	0	2021-12-19	8	8	8
9	29	9	0	0	0	2021-12-20	9	9	9
10	30	10	0	0	0	2021-12-21	10	10	10
11	31	11	0	0	0	2021-12-22	11	11	11
12	32	12	0	0	0	2021-12-23	12	12	12
13	33	13	0	0	0	2021-12-24	13	13	13
14	34	14	0	0	0	2021-12-25	14	14	14
15	35	15	0	0	0	2021-12-26	15	15	15
16	36	16	0	0	0	2021-12-27	16	16	16
17	37	17	0	0	0	2021-12-28	17	17	17

Vizualizare istoric bonuri

Adaugare bon

Modificare bon

Stergere bon

Delogare

Pe cel de al treilea panel acesta poate vizualiza bonurile fiscale introduse pana acum, dar poate adauga, modifica si sterge un anumit bon. La completarea casutei cu id-ul programarii se calculeaza automat casuta de la suma platita cu valoare corespunzatoare serviciilor efectuate.

Cel de-al patrulea panel este dedicat unei modalitati de programare mult mai usoare, ne mai fiind necesar sa tot accesam diferite tabele pentru a aduga o programare valida. Mai intal trebuie sa apasam vizualizare servicii, bifand cate un serviciu de langa si dand click in tabel pentru a alege serviciul dorit, completandu-se automat casutele din stanga. Dupa ce ne-am ales unul sau mai multe servicii (maxim 3) dand click pe vizualizare medici pentru acel serviciu putem vizualiza medicii care ar putea efectua serviciul dorit de pacient. Dand click pe un anumit medic se completeaza automat datele acestuia in stanga. Iar apoi trebuie dat click pe vizualizare zile cand medicul e la unitate pentru a vedea zile cand medicul e prezent in unitatea si cabinetul respectiv serviciilor. Alegand un rand din tabel se completeaza automat si data dorita, astfel ca mai ramane de adaugat doar id-pacient, ora_inceperii si ora_finalizarii. Dupa ce am completat toate campurile pentru a verifica daca programarea e corecta si a o adauga apasam pe verificare, astfel verificandu-se daca nu mai exista programari in acelasi timp, la acelasi doctor, cabinet si unitate, dar si daca diferenta dintre ora_inceperii si ora_finalizarii e egala cu timpul care trebuie alocat serviciilor alese. Daca toate datele sunt corecte se poate adauga programarea, altfel vor aparea mesaje corespunzatoare erorilor intampinate.

Politiclinică

Pacienti Programări Bon Fiscal Programare specială

Selecțai din tabele serviciul, unitatea și medicul:

Serviciul 1:

Serviciul 2:

Serviciul 3:

Echipament necesar:

Unitatea: Cabinetul:

Medic:

Data:

Introduceți manual aceste date:

ID programare:

ID pacient:

Ora începerii:

Ora finalizării:

După ce ați introdus datele, verificați dacă e valabilă programarea și apoi adăugați-o

Verificare programare

Adăugare programare

Id_servicii	Specialitate	Necesitate_compete..	Pret	Durata	echipament_necesar
1	cardiologie	specialist	100	60	ecograf
2	cardiologie	specialist	200	60	holter
3	neurologie	specialist	100	60	CT
4	pediatrie	specialist	100	60	Stetoscop si tensio...
5	ortopedie	specialist	100	60	punere ghips

☐ Serviciu 1 ☒ Serviciu 2 ☐ Serviciu 3

Vizualizează serviciile

id_progra...	id_pacient	id_servici...	id_servici...	id_servici...	ora_incep...	ora_finaliz...	datapacient	unitatea	cabinetul	medicul
1	21	1	0	0	8	9	2021-12-12	1	1	1
2	22	2	0	0	8	10	2022-12-13	2	2	2
3	25	5	0	0	9	10	2021-12-14	3	4	4
4	28	4	0	0	11	12	2021-12-15	4	5	7

Vizualizează program...

Vizualizeaza unități medicale

id_medice	nume	prenume	specialitate	grad	competente...	titlu_stiintific	echipamente	id_unitate	id_cabinet
2	Apostol	Maria	neurologie	specialist	explorare CT DA	CT	1	7	

Vizualizează medicii pentru acel serviciu

nume	prenume	specialitate	datapr	ora_incepere	ora_plecare	unitate	id_cabinet	id_ca_angajat
------	---------	--------------	--------	--------------	-------------	---------	------------	---------------

Vizualizează zilele când e medicul e la unitate

Delogare

Vaccinare

Policlinică - VACCINARE

— □ ×

Notă: Sunt acceptate modificări ulterioare.

Pentru a vă înregistra în vederea efectuării vaccinului "Pfizer" vă rugăm să completați câmpurile următoare:

ID:	<input type="text"/>
Nume:	<input type="text"/>
Prenume:	<input type="text"/>
Sunt de acord cu vaccinarea:	<input type="text" value="DA / NU"/>
Parolă:	<input type="password"/>

Trimite

Vă mulțumim pentru alegerea făcută!

Înapoi

Pe aceasta pagina utilizatorul isi poate introduce id-ul, numele, prenumele, acordul si parola de utilizator pentru a-si completa formularul referitor la sectiunea de vaccin.

Dupa ce apasa trimitere se completeaza automat acordul i tabelul utilizatori la coloana „vaccin”.

5. Securitate si justificarea solutiei alese

Am ales aceasta forma de implementare deoarece este accesibila atat utilizatorului cat si programatorului.

Este structurata pe module, fiecare modul are tipul de utilizatori care are acces si logarea prin id si parola.

Pentru cautare, inserare sau stergere (data, angajat, sau alte servicii) avem si optiunea de a afisa datele existente pentru a ne orienta si a gasi mai rapid ceea ce dorim. Si avand in vedere situatia actuala am avut grija sa pastram si o evidenta a persoanelor care doresc sa se vaccineze.

Pe partea financiara, ca dezvoltare ulterioara administratorul financiar ar putea modifica salariul oricarui angajat. Mai mult, acesta ar putea sa caute doar in functie de nume, adresa sau policlinica pentru a vedea pentru o singura persoana. De asemenea, un angajat ar putea sa vada pentru o singura luna ce salariu a avut, fara a mai cauta in istoricul salariilor.

O posibila dezvoltare ulterioara pentru modulul de resurse este adăugarea unui concediu medical "de urgență" programările fiind reatribuite altui angajat.

Ca securitate am utilizat frame-uri de login, in care utilizatorul trebuie sa isi introduca id-ul de utilizator si parola, astfel incat daca acesta introduce o parola gresita nu are acces la paginile urmatoare din aplicatie, altfel se vor deschide urmatoarele pagini in functie de pagina de pe care provine.